Theoretical Computer Science 412 (2011) 1338–1349

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

A graph-based framework for sub-pixel image segmentation

F. Malmberg [a,*], J. Lindblad [b], N. Sladoje [c], I. Nyström [a]

[a] *Centre for Image Analysis, Uppsala University, Sweden*

[b] *Centre for Image Analysis, Swedish University of Agricultural Sciences, Sweden*

[c] *Faculty of Technical Sciences, University of Novi Sad, Serbia*

ARTICLE INFO

Keywords:
Image segmentation
Graph labeling
Graph cuts
Coverage segmentation
Sub-pixel segmentation
Feature estimation

ABSTRACT

Many image segmentation methods utilize graph structures for representing images, where the flexibility and generality of the abstract structure is beneficial. By using a fuzzy object representation, i.e., allowing partial belongingness of elements to image objects, the unavoidable loss of information when representing continuous structures by finite sets is significantly reduced, enabling feature estimates with sub-pixel precision.

This work presents a framework for object representation based on fuzzy segmented graphs. Interpreting the edges as one-dimensional paths between the vertices of a graph, we extend the notion of a graph cut to that of a located cut, i.e., a cut with sub-edge precision. We describe a method for computing a located cut from a fuzzy segmentation of graph vertices. Further, the notion of vertex coverage segmentation is proposed as a graph theoretic equivalent to pixel coverage segmentations and a method for computing such a segmentation from a located cut is given. Utilizing the proposed framework, we demonstrate improved precision of area measurements of synthetic two-dimensional objects. We emphasize that although the experiments presented here are performed on two-dimensional images, the proposed framework is defined for general graphs and thus applicable to images of any dimension.

1. Introduction

Segmentation, the process of identifying and separating relevant objects and structures in an image, is a fundamental problem in image analysis. Accurate segmentation of objects of interest is essential for the success of many subsequent image analysis steps, e.g., estimation of geometrical features such as the area or perimeter of objects in the image. Even under the assumption that a correct segmentation is given, the accuracy of such measurements is still limited by the fact that we are trying to estimate features of continuous (real-world) objects based on a discrete, sampled, representation of the objects. This paper concerns a new framework for discrete object representation, with the aim of improving the precision and accuracy of feature estimates.

Several efficient methods for image segmentation have been formulated in the framework of *edge weighted graphs*. The graph theoretic approach to image processing naturally leads to methods that are applicable to images of any dimension, and images sampled on non-Cartesian or spatially variant grids [1,2]. An image may be associated with a graph by identifying each image element with a vertex in the graph, and defining an edge set to represent local adjacency between image elements. Each edge in the graph is also associated with a (real-valued) *weight*, reflecting the image content [3]. A segmentation of a graph may be formulated either as a mapping from the vertices of the graph to some set of object

* Corresponding address: Centre for Image Analysis, Uppsala University, SE-751 05 Uppsala, Sweden. Tel.: +46 0 18 471 78 49.
E-mail address: filip@cb.uu.se (F. Malmberg).

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.11.030

categories, or as a *graph cut*. Informally, a cut is a set of edges such that if they are removed, the graph is separated into two or more connected components. The two representations are closely related, and the choice of one representation over the other is largely a matter of preference.

A segmentation is *crisp* if each element is associated with exactly one object category. In contrast, a *fuzzy* segmentation allows *partial* belongingness to several object categories. By allowing image elements to (partly) belong to more than one object class, the loss of information associated with the process of image segmentation can be significantly reduced. Even though features such as area or perimeter are defined for general fuzzy sets (see, e.g., [4]), it is important to keep in mind that fuzzy memberships assigned by a particular membership function may reflect various properties of objects and pixels, depending on what the segmentation method was based on. However, to have meaningful area or perimeter estimates from fuzzy memberships, fuzzy membership values must reflect some geometrical properties of objects and pixels in their discrete representations. In our previous work, we explored discrete object representations where the values assigned to an image element are proportional to the (relative) coverage of that image element by each of the imaged objects. Clearly, such a *pixel coverage* based membership function is suitable for the task of estimating geometric features of objects. Previous studies have confirmed that this class of fuzzy representations enables geometric features estimation with significantly increased precision, compared to crisp representations, see, e.g., [5–7].

Here, we present a framework for extending the concept of fuzzy segmentation and partial belongingness to segmentation on graphs. Commonly, a segmentation is only defined at the vertices of a graph. Here, we interpret the edges of the graph as paths between the vertices. This allows us to define points along the edges of the graph, and to assign each such point a belongingness to one or more object classes. Thereby, we obtain an *edge segmentation* of the graph. These concepts are introduced in Section 3. In Section 4, we introduce the concept of *located cuts*, which are graph cuts specified with sub-edge precision. Via the concept of *induced edge segmentation*, located cuts provide a convenient way of extending a segmentation defined on the vertices of the graph to all points along the edges of the graph. Furthermore, we define *vertex coverage segmentation* as a graph theoretic equivalent of pixel coverage segmentation. Computing a vertex coverage segmentation involves integrating over all points along the edges of the graph. We show that for segmentations derived from located cuts, the involved integrals may be reduced to simple closed formulas that are easy to evaluate.

In Section 5, we present a practical method for computing a located cut, starting from an arbitrary fuzzy segmentation.

To illustrate the benefits of the proposed framework, we perform an empirical study where we measure the area of synthetic two-dimensional (2D) objects. The results of this study show that the proposed vertex coverage representation allows area to be measured with much higher precision, compared to crisp representations. It is reasonable to expect that similar improvements can be achieved for other features by following the same approach.

This paper extends our previous work, presented in [8], in the following ways:

- We provide a formal framework for the proposed approach, thereby increasing its generality and applicability.
- We propose a general method for computing located cuts, starting from an arbitrary fuzzy segmentation of the vertices of the graph. The proposed method is not tied to a particular segmentation method, but may be used in conjunction with a number of popular graph-based segmentation methods.
- We evaluate the framework by measuring the area of a large number of synthetic 2D objects, comparing traditional crisp object representation with the proposed vertex coverage representation. Significant improvements in measurement precision are observed.

2. Background

In this section, we first present a number of graph-based segmentation methods that can be used as input to the described framework. Thereafter, we discuss the concept of image elements being partially covered by one or several objects.

2.1. Graph-based segmentation methods

Graph-based methods have been particularly popular in the context of *seeded* segmentation. Seeded segmentation methods attempt to solve the segmentation problem in the presence of prior knowledge in the form of a partial segmentation. Given an image where a small subset of the image elements (called *seed-points*) have been assigned segmentation labels (e.g., object or background), an automatic method completes the labeling for all image elements. The seed-points may be provided either by some pre-processing algorithm, or by a human user in an interactive setting. Here, we review a selection of graph-based methods for seeded segmentation. It should be noted that the framework proposed here is not tied to any particular segmentation method. With limited work, any of the segmentation methods presented below can be adapted to our framework, thus allowing improved feature measurements.

The *minimal graph cuts* [9] method calculates a cut separating the seed-points, such that the sum of the edge weights along the cut is minimal. A variant of this method is the *normalized cuts* algorithm [10]. Another family of methods is based on the calculation of a *minimum cost path forest*. These methods calculate a cut such that each vertex is connected to the closest seed-point, as determined by some *path cost function*. Examples of this approach include the *Image Foresting Transform* (IFT) [11,12], and the *Relative Fuzzy Connectedness* method [13]. Malmberg et al. [14] recently proposed the *Relaxed* IFT, that extends the IFT by enforcing smoothness of the segmentation boundary. The *Random Walker* [15] method computes cuts

such that each vertex is connected to the seed-point that a "random walker", starting at the vertex, is expected to reach first. Other segmentation approaches that have been successfully re-formulated on weighted graphs include the classical watershed approach [16] and segmentation based on minimizing the Mumford-Shah functional [17].

Many of the above methods are closely related, and several efforts have been made to clarify the theoretical relation between the methods. A unifying framework for seeded segmentation was presented by Sinop and Grady [18], and extended by Couprie et al. [19]. In [20], Miranda and Falcão established a link between segmentation based on minimum cost paths and the minimal graph cuts approach.

2.2. Pixel coverage segmentation

Image elements partly covered by more than one object are often referred to as mixed pixels. It is not surprising that the issue of mixed pixels is thoroughly addressed in remote sensing applications (see, e.g., [21–23]). Pixels in remotely sensed images are of sizes that very often lead to individual pixels being covered by several classes/objects imaged on the ground. Sub-pixel proportion estimation, leading to so-called fraction images, is a widely studied problem. Commonly used approaches are linear mixture models and neural networks, the former due to their simplicity, and the latter when the mixtures in the pixels are non-linear and more complex [22].

Interest for sub-pixel segmentation approaches exists in cases of images of higher resolution as well. In particular, three-dimensional (3D) medical image analysis often requires treatment of voxels that suffer from partial volume effects, i.e., the voxels contain a mixture of two or more tissue types [24]. The issue is particularly studied for magnetic resonance (MR) and positron emission tomography (PET) images of the human brain. The importance of sub-voxel measurements is convincingly demonstrated in Leemput et al. [24], where it is shown that consistently misplacing the tissue borders, in a brain volume having voxels of size 1 mm^3, by only one voxel resulted in volume errors of approximately 30%, 40%, and 60% for white matter, grey matter, and cerebrospinal fluid (CSF), respectively. A number of methods have been suggested to overcome this limitation, and to address the issue of partial volume effect corrections. Common methods are based on expectation-maximization [24], scale-space approaches [25], wavelets [26], Markov random fields [27], fuzzy techniques [28], etc.

3. Notation and definitions

In this section, we present basic definitions for graphs, graph cuts, and segmentation of graph vertices. Moreover, we introduce the novel concept of *edge segmentation* of a graph.

3.1. Graphs and graph cuts

A *graph* is here defined as a pair $G = (V(G), E(G))$, consisting of vertices $v \in V(G)$ and edges $e \in E(G) \subseteq V \times V$. In order to simplify the notation, the vertices and edges of a graph will be denoted V and E instead of $V(G)$ and $E(G)$ whenever it is clear from the context which graph G they belong to. An edge spanning two vertices v_i and v_j is denoted e_{ij}. If $e_{ij} \in E$, the vertices v_i and v_j are *adjacent*. The set of vertices adjacent to a vertex v is denoted by $\mathcal{N}(v)$. The following will assume that the graph is undirected, i.e., $e_{ij} \in E \Leftrightarrow e_{ji} \in E$.

A *path* is an ordered sequence of vertices $\pi = \langle v_1, v_2, \ldots, v_k \rangle$ such that $v_{i+1} \in \mathcal{N}(v_i)$ for all $i \in [1, k-1]$. We denote the origin v_1 and destination v_k of π by $\mathrm{org}(\pi)$ and $\mathrm{dst}(\pi)$, respectively. Two vertices v and w are *linked* in G if there exists a path π in G such that $\mathrm{org}(\pi) = v_i$ and $\mathrm{dst}(\pi) = w$. The notation $v_i \underset{G}{\sim} v_j$ will here be used to indicate that v_i and v_j are linked on G. If all pairs of vertices in G are linked, then G is *connected*, otherwise it is *disconnected*.

If G and H are graphs such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ then H is a *subgraph* of G. If H is a connected subgraph of G and $v_i \underset{G}{\not\sim} v_j$ for all $v_i \in H$ and $v_j \notin H$, then H is a *connected component* of G.

Definition 1 (*Graph Cut*)**.** Let $G = (V, E), S \subseteq E$, and $G' = (V, E \setminus S)$. If, for all $e_{ij} \in S$, it holds that $v_i \underset{G'}{\not\sim} v_j$, then S is a (graph) cut on G.

For any cut $S \neq \emptyset$, the graph $(V, E \setminus S)$ is disconnected, i.e., it consists of two or more connected components.

3.2. Graph segmentation

A *segmentation* of a graph into k object classes is here defined as a function that maps elements of the graph (vertices and edges) to vectors $x = (x_1, x_2, \ldots, x_k) \in \mathbb{R}^k$ such that

$$x_i \geq 0 \quad \text{for all } i \in \{1, 2, \ldots, k\} \tag{1}$$

and

$$\|x\|_1 = 1. \tag{2}$$

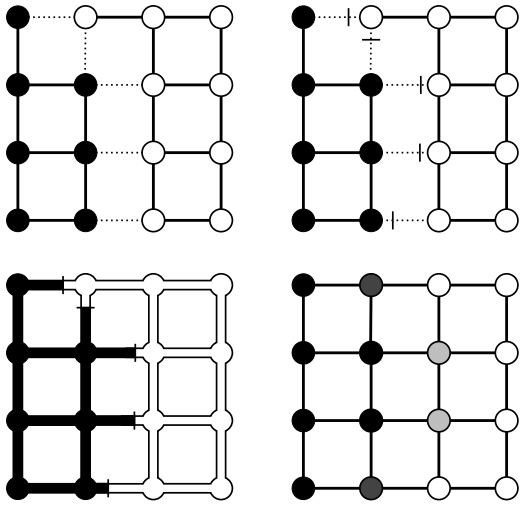

Fig. 1. Components of the proposed framework. (Top left) A crisp vertex segmentation $\mathcal{V}$ of a graph. The boundary, $\partial\mathcal{V}$, of the segmentation is shown as dashed lines. (Top right) A corresponding located cut. (Bottom left) The edge segmentation $\mathcal{I}_{\mathcal{V},\mathcal{T}}$ induced by $\mathcal{V}$ and $\mathcal{T}$. (Bottom right) One component of the corresponding vertex coverage segmentation $\mathcal{C}_{\mathcal{V},\mathcal{T}}$.

The set of vectors that satisfy Eqs. (1) and (2) is denoted U^k. Each component x_i in x represents the degree to which the graph element belongs to the corresponding class. If all $x_i \in \{0, 1\}$ then x is *crisp*, otherwise it is *fuzzy* [29].

A *vertex segmentation* assigns a vector from U^k to each vertex of the graph.

Definition 2 (*Vertex Segmentation*)**.** Let $G = (V, E)$. A vertex segmentation $\mathcal{V}$ of G is a mapping $\mathcal{V} : V \to U^k$.

Vertex segmentations and graph cuts are closely related. The *boundary*, $\partial\mathcal{V}$, of a vertex segmentation $\mathcal{V}$ is here defined as the edge set $\partial\mathcal{V} = \{e_{ij} \in E \mid \mathcal{V}(v_i) \neq \mathcal{V}(v_j)\}$. The relation between segmentations and cuts is summarized in the following, well-known, theorem.

Theorem 1. *For any graph $G = (V, E)$ and set of edges $S \subseteq E$, the following statements are equivalent:*

1. *There exists a vertex segmentation $\mathcal{V}$ of G such that $S = \partial\mathcal{V}$.*
2. *S is a cut on G.*

Proof. First, let $\mathcal{V}$ be a vertex segmentation of G and let $G' = (V, E \setminus \partial\mathcal{V})$. For any path π on G' it holds that $\mathcal{V}(\text{org}(\pi)) = \mathcal{V}(\text{dst}(\pi))$. For all $e_{ij} \in \partial\mathcal{V}$, however, it holds that $\mathcal{V}(v_i) \neq \mathcal{V}(v_j)$ and so $v_i \not\underset{G'}{\sim} v_j$. Thus, $S = \partial\mathcal{V}$ is a cut on G.

Next, let $S \subseteq E$ be a cut on G and let $G' = (V, E \setminus S)$. Let $\mathcal{V}$ be a vertex segmentation of G such that each connected component of G' has a unique label. Then for any edge $e_{kl} \in E$ it holds that $\mathcal{V}(v_k) = \mathcal{V}(v_l) \Leftrightarrow v_k \underset{G'}{\sim} v_l$, and thus $\partial\mathcal{V} = S$. □

Throughout this paper, we interpret edges as paths between the vertices of a graph, and thus a point along an edge may be specified by a parameter $t \in [0, 1]$. A point on a graph G is a pair (e, t), where $e \in E$ and $t \in [0, 1]$. In particular, points corresponding to vertices are of the form $(e, 0)$ or $(e, 1)$. An *edge segmentation* assigns a vector from U^k to every point along the edges of the graph.

Definition 3 (*Edge Segmentation*)**.** Let $G = (V, E)$. An edge segmentation $\mathcal{E}$ of G is a mapping $\mathcal{E} : E \times [0, 1] \to U^k$.

For an undirected graph, we require that

$$\mathcal{E}(e_{ij}, t) = \mathcal{E}(e_{ji}, 1 - t) \tag{3}$$

for all $e_{ij} \in E$ and $t \in [0, 1]$. A vertex segmentation $\mathcal{V}$ and an edge segmentation $\mathcal{E}$ are said to be *consistent* if $\mathcal{V}(v_i) = \mathcal{E}(e_{ij}, 0)$ for all $e_{ij} \in E$.

4. A framework for sub-pixel segmentation on graphs

In this section, we introduce the proposed framework for sub-pixel segmentation on graphs. The components of the proposed framework are illustrated in Fig. 1.

4.1. Located cuts

As established in Section 3.2, the boundary of a vertex segmentation is a cut that separates objects represented by the graph. In that separation, edges that belong to the cut are not assigned to any of the objects. To increase the precision of

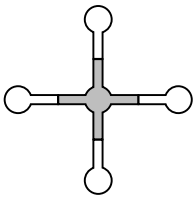

Fig. 2. The domain (shown in gray) of a vertex with four neighbors.

this separation, we suggest to consider the edges of the cut as well, and specify a point along each edge of the cut where the transition between different objects occur. A cut specified in this way is called a *located cut*.

Definition 4 (*Located Graph Cut*)**.** Let $G = (V, E)$ be an undirected graph. A located (graph) cut on G is a pair $(S, \mathcal{T})$, where $S \subseteq E$ is a cut on G and $\mathcal{T}$ is a mapping $\mathcal{T} : S \to [0, 1]$ such that $\mathcal{T}(e_{ij}) = 1 - \mathcal{T}(e_{ji})$ for all $e_{ij} \in S$. We say that $\mathcal{T}$ is the *location* of S.

Located cuts provide a natural way to define an edge segmentation based on a vertex segmentation, via the concept of *induced edge segmentation*.

Definition 5 (*Induced Edge Segmentation*)**.** Given a vertex segmentation $\mathcal{V}$, and location $\mathcal{T}$ such that $(\partial\mathcal{V}, \mathcal{T})$ is a located cut, the induced edge segmentation $\mathcal{I}_{\mathcal{V},\mathcal{T}}$ is defined as

$$\mathcal{I}_{\mathcal{V},\mathcal{T}}(e_{ij}, t) = \begin{cases} \mathcal{V}(v_i) & \text{if } e_{ij} \in \partial\mathcal{V} \text{ and } t < \mathcal{T}(e_{ij}) \\ \frac{1}{2}(\mathcal{V}(v_i) + \mathcal{V}(v_j)) & \text{if } e_{ij} \in \partial\mathcal{V} \text{ and } t = \mathcal{T}(e_{ij}) \\ \mathcal{V}(v_j) & \text{otherwise.} \end{cases} \tag{4}$$

The induced edge segmentation $\mathcal{I}_{\mathcal{V},\mathcal{T}}$ is consistent with $\mathcal{V}$. The practical benefit of the definition of induced edge segmentation is that given a vertex segmentation $\mathcal{V}$, specifying a consistent edge segmentation is reduced to defining $\mathcal{T}$. The issue of defining $\mathcal{T}$ is discussed in Section 5. In the remainder of this section, we assume that $\mathcal{T}$ is given.

4.2. Vertex coverage segmentation

If $\mathcal{V}$ is a vertex segmentation and $\mathcal{E}$ is an edge segmentation such that $\mathcal{V}$ and $\mathcal{E}$ are consistent, then we may view $\mathcal{E}$ as an extension of $\mathcal{V}$ from the set of points for which $t = 0$ (points corresponding to vertices) to *all* points along the edges of the graph. In this sense, $\mathcal{E}$ contains more information than $\mathcal{V}$. In order to obtain good feature measurements of segmented objects on the graph, we would like to take advantage of this additional information. Existing feature estimators, however, are defined for vertex segmentations only. It is therefore of high practical interest to define a way of converting an edge segmentation to a vertex segmentation, while preserving as much of the information from the edge segmentation as possible. For this purpose, we now introduce the concept of *vertex coverage segmentation*, a graph theoretic equivalent of the concept of pixel coverage segmentation discussed in Section 2.2.

We define the *domain* of a vertex v_i as the set of points on the "half-edges" adjacent to the vertex, see Fig. 2. Let $\mathcal{E}$ be an edge segmentation of G. The vertex coverage segmentation $\mathcal{C}_{\mathcal{E}}$ of a vertex v_i is a vector of U^k defined as

$$\mathcal{C}_{\mathcal{E}}(v_i) = \frac{\sum_{j, v_j \in \mathcal{N}(v_i)} 2\int_0^{\frac{1}{2}} \mathcal{E}(e_{ij}, t)\mathrm{d}t}{|\mathcal{N}(v_i)|} \tag{5}$$

for all $v_i \in V$.

For an arbitrary edge segmentation, the integral in the numerator of Eq. (5) may not be possible to evaluate analytically. However, for an induced edge segmentation $\mathcal{I}_{\mathcal{V},\mathcal{T}}$, the integral can be written in closed form as

$$2\int_0^{\frac{1}{2}} \mathcal{I}_{\mathcal{V},\mathcal{T}}(e_{ij}, t)\mathrm{d}t = \begin{cases} 2\mathcal{T}(e_{ij})\mathcal{V}(v_i) + (1 - 2\mathcal{T}(e_{ij}))\mathcal{V}(v_j) & \text{if } \mathcal{T}(e_{ij}) < \frac{1}{2} \\ \mathcal{V}(v_i) & \text{otherwise.} \end{cases} \tag{6}$$

To simplify the notation, a vertex coverage segmentation of an induced edge segmentation $\mathcal{I}_{\mathcal{V},\mathcal{T}}$ will be denoted $\mathcal{C}_{\mathcal{V},\mathcal{T}}$.

5. Computing located cuts

In this section, we consider the issue of computing located cuts in practice. Many graph-based segmentation methods produce fuzzy vertex segmentations, with fuzzy memberships assigned based on various criteria. Even though such a representation contains more information about (some properties of) the original, this information is often not utilized in further processing steps, but a defuzzification to a crisp vertex segmentation is instead performed [15,18,19,14].

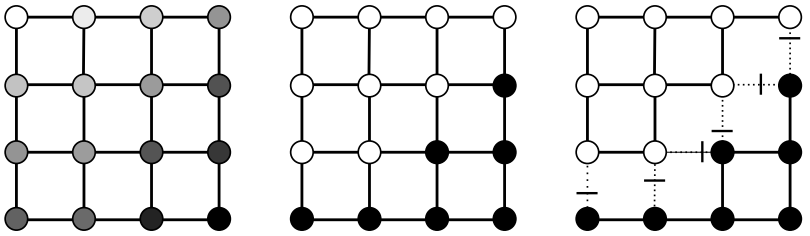

Fig. 3. Computing located cuts in the context of defuzzification. (Left) One component of a fuzzy vertex segmentation $\mathcal{V}$ of a graph. (Middle) Corresponding defuzzified vertex segmentation $\hat{\mathcal{V}}$. (Right) A located cut $(\partial\hat{\mathcal{V}}, \mathcal{T})$, obtained by combining information from $\mathcal{V}$ and $\hat{\mathcal{V}}$.

Our previous work [30,5,7,6] showed that significant improvements in precision of feature estimates can be achieved by appropriately utilizing the information contained in a fuzzy representation, and in particular that, for the specific case of pixel coverage representations, mathematical results are derived [5,7] showing how the use of coverage information can overcome insufficient spatial resolution and provide estimates with sub-pixel precision. Inspired by these results, we developed in [8] a method for computing an approximate vertex coverage segmentation, based on the IFT. That method, utilizing a graph framework, and essentially computing a located cut,[1] can be seen as one special case of the framework presented in this work.

In this section, we present a general method for computing a located cut starting from an *arbitrary* fuzzy vertex segmentation. The source of the segmentation is not vital, it may be an IFT based segmentation, as in [8], or some other segmentation method. For example, in the evaluation section of this paper we use a signed distance transform to directly generate a fuzzy vertex segmentation from (continuous) geometric objects.

It should be pointed out that geometric features can be computed directly from a fuzzy vertex segmentation. However, the meaning of these extracted feature values will depend on the particular meaning of the membership function used. The scheme presented in the following provides approximate vertex coverage values derived from a given fuzzy segmentation. In that sense, the presented method is a means for transforming one fuzzy vertex segmentation into another one (a vertex coverage segmentation), where the latter one has a well defined interpretation of its membership values, and is shown to be most suitable for feature computation of high precision.

The method starts with an initial defuzzification step, defining the approximate boundaries of the image objects. Utilizing the information from the fuzzy vertex segmentation, a located cut for the boundary of the defuzzified vertex segmentation is then derived. The information contained in the fuzzy vertex segmentation can in this way be utilized to define an induced edge segmentation with sub-edge precision. The process is summarized in Fig. 3, and described in detail in the following.

In the context of seeded segmentation, defuzzification is commonly performed by associating each vertex with the object for which the degree of membership is maximum. The maximum element of the segmentation vector is not necessarily unique, and therefore this defuzzification strategy may give rise to ambiguities. If a strictly crisp segmentation is required, then these ambiguities need to be resolved using some additional criterion. Here, we choose to assign the mean of the maximum elements, thereby producing an *almost* crisp segmentation. Given a vector $x \in \mathbb{R}^k$, we define the defuzzified vector $\hat{x} \in U^k$ as

$$\hat{x} = \frac{\mathcal{M}(x)}{\|\mathcal{M}(x)\|_1}, \tag{7}$$

where $\mathcal{M} : \mathbb{R}^k \to \{0, 1\}^k$ is defined as

$$\mathcal{M}(x)_i = \begin{cases} 1 & \text{if } x_i = \|x\|_\infty \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

If x has a unique maximum element, then $\hat{x}$ is crisp. For a vertex segmentation $\mathcal{V}$, the defuzzified vertex segmentation $\hat{\mathcal{V}}$ is defined as

$$\hat{\mathcal{V}}(v) = \widehat{\mathcal{V}(v)}, \tag{9}$$

for all $v \in V$.

We now use information from $\mathcal{V}$ to compute a location $\mathcal{T}$ such that $(\partial\hat{\mathcal{V}}, \mathcal{T})$ is a located cut. From the definition of $\hat{\mathcal{V}}$, we know that for each edge $e_{ij} \in \partial\hat{\mathcal{V}}$, the set of maximal elements of $\mathcal{V}(v_i)$ is different from the set of maximal elements of $\mathcal{V}(v_j)$. Our strategy for determining a located cut is to perform a linear interpolation of the values of $\mathcal{V}$ along the edges in $\partial\hat{\mathcal{V}}$, to find the point where the maximal element changes. For each edge $e_{ij} \in \partial\hat{\mathcal{V}}$, we define the following scalar values:

$$\begin{aligned} a &= \mathcal{V}(v_i) \cdot \hat{\mathcal{V}}(v_i) \\ b &= \mathcal{V}(v_j) \cdot \hat{\mathcal{V}}(v_i) \\ c &= \mathcal{V}(v_j) \cdot \hat{\mathcal{V}}(v_j) \\ d &= \mathcal{V}(v_i) \cdot \hat{\mathcal{V}}(v_j). \end{aligned} \tag{10}$$

[1] Although the terminology "located cut" was not used in [8].

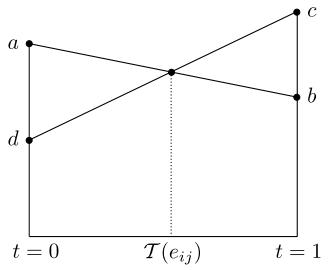

Fig. 4. Finding the point $\mathcal{T}(e_{ij})$ by linear interpolation.

The vector $\hat{\mathcal{V}}(v_i)$ corresponds to an (almost) crisp segmentation of the object at v_i. The values a and b represent the degree to which this object is present in the fuzzy segmentation $\mathcal{V}$ at v_i and v_j, respectively. In the same way, $\hat{\mathcal{V}}(v_j)$ corresponds to an (almost) crisp segmentation of the object at v_j, and the values c and d represent the degree to which this object is present in the fuzzy segmentation $\mathcal{V}$ at v_j and v_i. Interpolating these values linearly over the edge, we obtain the following linear equation for the point $\mathcal{T}(e_{ij})$:

$$(1 - \mathcal{T}(e_{ij}))a + \mathcal{T}(e_{ij})b = \mathcal{T}(e_{ij})c + (1 - \mathcal{T}(e_{ij}))d. \tag{11}$$

See Fig. 4. Solving Eq. (11) for $\mathcal{T}(e_{ij})$, we obtain

$$\mathcal{T}(e_{ij}) = \frac{a - d}{a - d + c - b}. \tag{12}$$

In the remainder of this section, we show that for the location $\mathcal{T}$ defined by Eq. (12), $(\partial\hat{\mathcal{V}}, \mathcal{T})$ is a valid located cut.

Lemma 1. *Let $x, y \in \mathbb{R}^k$. Then $x \cdot \hat{x} \geq x \cdot \hat{y}$.*

Proof. For any vector $z \in U^k$, it holds that $x \cdot z \leq \|x\|_\infty$. In particular, this holds for $\hat{y}$. It follows from the definition of $\hat{x}$ that $\hat{x}_i > 0 \Rightarrow x_i = \|x\|_\infty$ for all $i \in \{1, 2, \ldots, k\}$. Therefore,

$$x \cdot \hat{x} = \|\hat{x}\|_1 \|x\|_\infty = \|x\|_\infty. \tag{13}$$

Thus, $x \cdot \hat{x} \geq x \cdot \hat{y}$. □

Lemma 2. *Let $x, y \in \mathbb{R}^k$. If*

$$x \cdot \hat{x} = x \cdot \hat{y} \tag{14}$$

and

$$y \cdot \hat{y} = y \cdot \hat{x}, \tag{15}$$

then $\hat{x} = \hat{y}$.

Proof. If Eq. (14) holds, then $y_i = \|y\|_\infty \Rightarrow x_i = \|x\|_\infty$ for $i \in \{1, 2, \ldots, k\}$. If Eq. (15) holds, then $x_i = \|x\|_\infty \Rightarrow y_i = \|y\|_\infty$. Thus, if both Eqs. (14) and (15) hold, then $x_i = \|x\|_\infty \Leftrightarrow y_i = \|y\|_\infty$, and so $\hat{x} = \hat{y}$. □

Theorem 2. *For all $e_{ij} \in \partial\hat{\mathcal{V}}$, $\mathcal{T}(e_{ij})$ as defined in Eq. (12) exists and $\mathcal{T}(e_{ij}) \in [0, 1]$.*

Proof. From Lemma 1 it follows that $a \geq d$ and $c \geq b$. From Lemma 2 it follows that on the boundary of $\hat{\mathcal{V}}$, at least one of these inequalities must be strict. Thus for all $e_{ij} \in \partial\hat{\mathcal{V}}$, it holds that $a - d + c - b > 0$ and $a - d \leq a - d + c - b$. □

6. Evaluation

To study the ability of the proposed framework to produce segmentations from which features can be measured with improved precision, we perform an empirical study where we measure the area of analytic 2D objects.

6.1. Synthetic data

Three sets of analytic objects are used in the experiments. The first set consists of 200 Euclidean disks, with radii evenly distributed in the range 10–50 pixels. The second set consists of 200 squares, rotated with angles evenly distributed between 0 and 90 degrees. The third set consists of 200 axis-aligned Koch snowflakes with two levels of subdivision and with areas evenly distributed in the range 500–9000 pixels. For each of the 600 objects in the experiment, the center of the object is

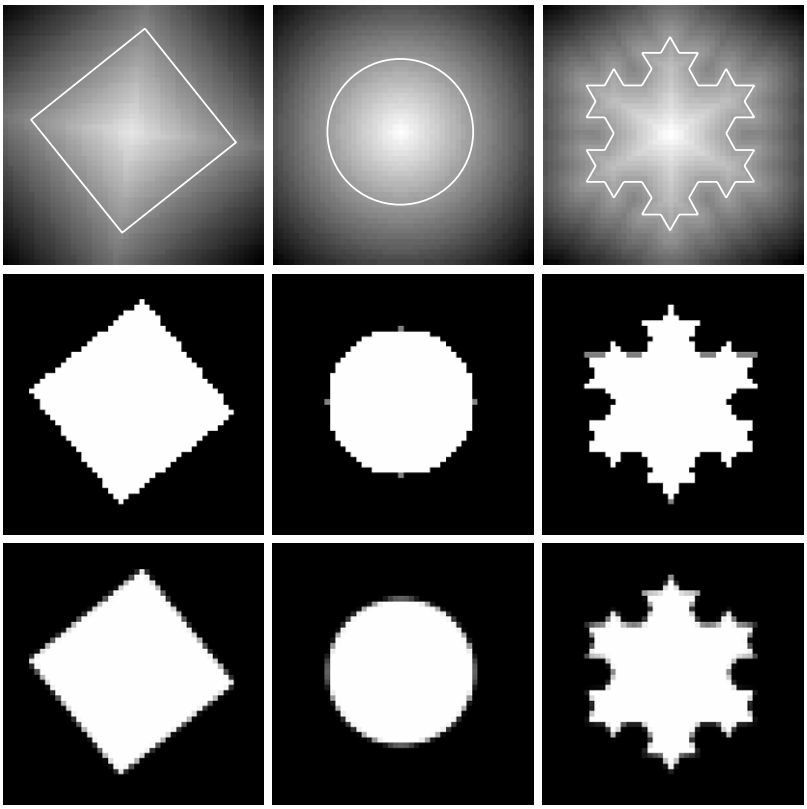

Fig. 5. Synthetic objects used in the experiment. (Top row) First component x_1 of the fuzzy representations $\mathcal{V}$ of the analytic objects. Boundaries of the corresponding analytic objects are superimposed in white. (Middle row) Defuzzified objects $\hat{\mathcal{V}}$. (Bottom row) Corresponding vertex coverage segmentations $\mathcal{C}_{\hat{\mathcal{V}},\mathcal{T}}$.

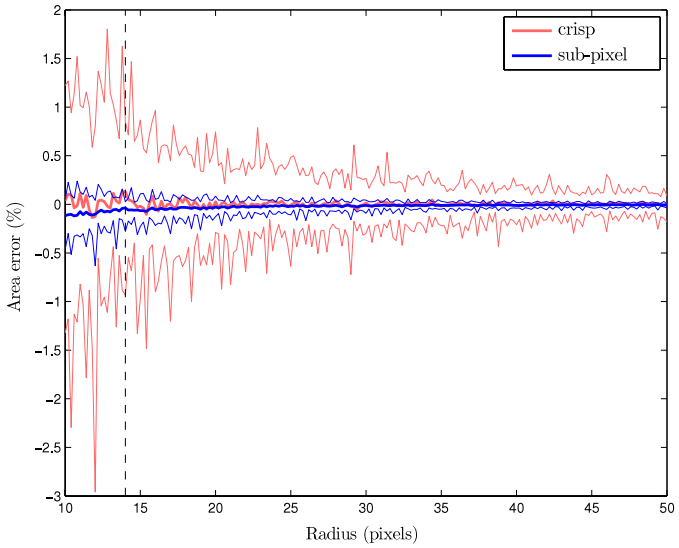

Fig. 6. Relative error of the area measurements for the Euclidean disks. The curves show the maximum, minimum, and mean relative error for each object with respect to shifting the center point of the object within a pixel. The dashed vertical line indicates the radius of the disk shown in Fig. 5.

shifted to 50 random positions within one pixel, and the digitization and area measurement procedure described below is repeated for every position. Thus, a total of 30 000 measurements are performed.

Throughout the experiments, an 8-connected lattice is used to define the image graph. Fuzzy vertex segmentations $\mathcal{V}(v) = (x_1, 1 - x_1)$ of the synthetic objects are defined according to a signed distance map (corresponding to a level-set representation) normalized to the range $[0, 1]$. For disks and squares, the signed distance transform is computed directly

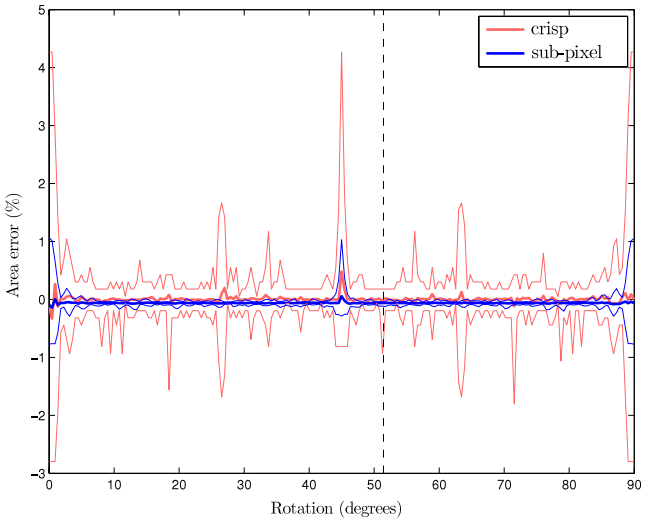

Fig. 7. Relative error of the area measurements for the rotated squares. The curves show the maximum, minimum, and mean relative error for each object with respect to shifting the center point of the object within a pixel. The dashed vertical line indicates the rotation of the square shown in Fig. 5.

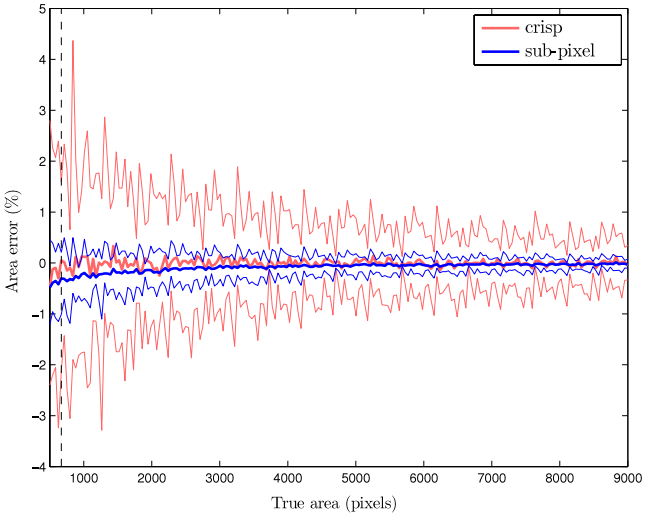

Fig. 8. Relative error of the area measurements for the Koch snowflakes. The curves show the maximum, minimum, and mean relative error for each object with respect to shifting the center point of the object within a pixel. The dashed vertical line indicates the size of the snowflake shown in Fig. 5.

from their mathematical definition. The signed distance transform of a Koch snowflake is calculated in two steps. First, we calculate the minimum distance between each pixel and the boundary of the snowflake. The boundary consists of a finite number of line segments, and the minimum distance between a point and a line segment is straightforward to calculate analytically. Next, we we rasterize all the constituent triangles of the snowflake to determine the pixels that are inside the object, and multiply the distance value of these pixels by -1 to obtain the signed distance transform. Fig. 5 (top row) presents the first component x_1 of $\mathcal{V}$ with the boundaries of the synthetic objects superimposed.

6.2. Area estimation

The area (in pixels) of the object corresponding to the ith component of a vertex segmentation $\mathcal{V}'$ is measured by summing the values of that component for all vertices in the graph:

$$A(\mathcal{V}', i) = \sum_{v \in V} \mathcal{V}'(v)_i. \tag{16}$$

For the fuzzy segmentation $\mathcal{V}$, the membership value of a pixel is related to the distance to the boundary of the continuous object. Computing the area of $\mathcal{V}$ does not provide a sensible estimate of the area of the continuous object, since the required geometric information is only indirectly contained in the assigned pixel membership values. For each fuzzy segmentation $\mathcal{V}$, we therefore compute a defuzzified vertex segmentation $\hat{\mathcal{V}}$ and a corresponding location $\mathcal{T}$ using the method presented

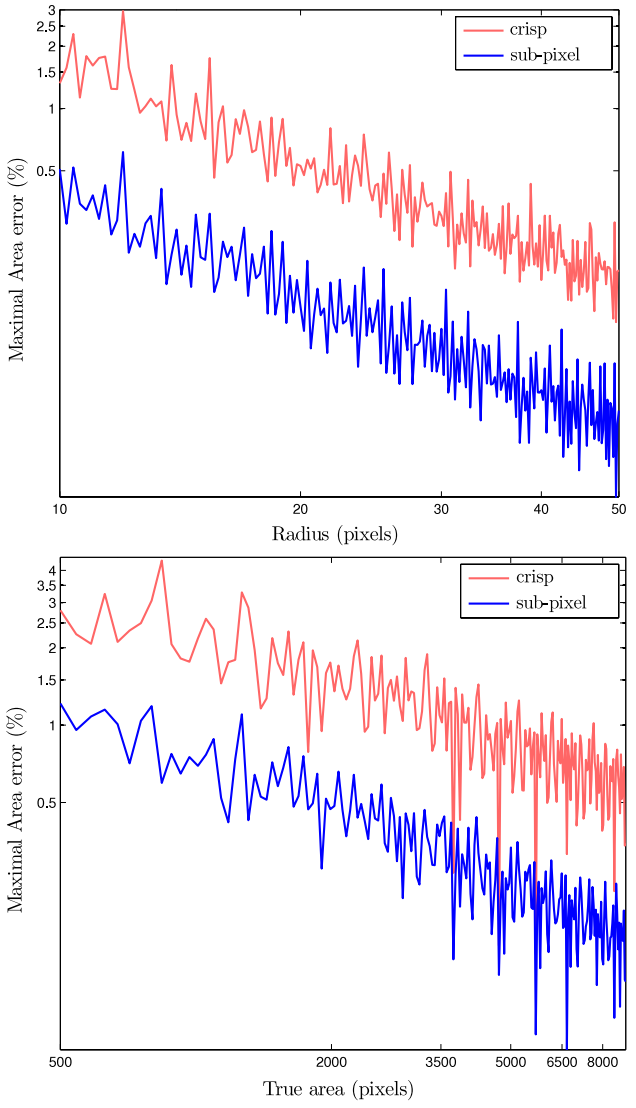

Fig. 9. Maximal relative error, with respect to centre point shifts, of the area measurements for disks (top) and Koch snowflakes (bottom) plotted in log–log scale. The convergence rate is the same for measurements made from crisp segmentations and vertex coverage segmentations, while the magnitude of the error is smaller for the vertex coverage segmentations.

in Section 5. In addition, we compute the vertex coverage segmentation $\mathcal{C}_{\hat{\mathcal{V}},\mathcal{T}}$. The segmentations $\hat{\mathcal{V}}$ and $\mathcal{C}_{\hat{\mathcal{V}},\mathcal{T}}$ are illustrated in the middle and bottom row of Fig. 5, respectively. We compare area measurements made from the almost crisp vertex segmentations $\hat{\mathcal{V}}$ with area measurements made from the vertex coverage segmentations $\mathcal{C}_{\hat{\mathcal{V}},\mathcal{T}}$.

Area estimation based on counting the number of object pixels in a crisp segmentation in $\mathbb{Z}^2$ has been shown to be multigrid convergent for non-fractal objects [31], i.e., the relative estimation error decreases with increase of the image resolution, or, alternatively, size of an object. If the observed graph is embedded in $\mathbb{Z}^2$, and used for representation of an image object, the area estimates obtained from a vertex segmentation $\mathcal{C}_{\hat{\mathcal{V}},\mathcal{T}}$, as well as from $\hat{\mathcal{V}}$, by summing the values assigned to vertices, are also multigrid convergent. This follows from the fact that for both these representations, only the values of boundary vertices are different from what is obtained from a crisp representation of the same object. The number of boundary vertices are too few ($\mathcal{O}(\sqrt{|V|})$, where $|V|$ is the number of graph vertices) compared to the total area estimate ($\mathcal{O}(|V|)$) to affect the convergence that holds for the crisp case. Without further knowledge of the geometry of the graph, however, it does not seem possible to improve on the theoretical convergence of the crisp case ($\mathcal{O}(\frac{1}{\sqrt{|V|}})$). This is highlighted in Fig. 9, which shows a log–log plot of the measurement errors for disks and snowflakes. The slope of the curves in Fig. 9 is the same for measurements based on the almost crisp segmentations and measurements based on the vertex coverage segmentations, indicating that the convergence rate is the same.

Even though the convergence rate is the same, the area estimation error obtained by the proposed method is, for every given resolution, significantly smaller than the error obtained from a crisp representation. This is clearly visible in Figs. 6–8.

Fig. 10. A kidney, segmented from an MR volume image of a human abdomen using the Relaxed IFT method [14]. (Top) Original volume image. (Middle) Crisp segmentation. (Bottom) Vertex coverage segmentation.

The plots show the maximum, mean, and minimum relative error of the measured area with respect to center point shifts for disks, squares, and snowflakes, respectively. The precision of the feature measurements is greatly improved for the vertex coverage segmentations $\mathcal{C}_{\hat{\mathcal{V}},\mathcal{T}}$, compared to the almost crisp $\hat{\mathcal{V}}$ segmentations. The standard deviation of the relative error, with respect to center point shifts, is on average reduced by a factor 4.7 for all objects in the experiment. The measurements based on the vertex coverage segmentations suffer from a small systematic under-estimation, compared to the analytic value. This bias is likely caused by the linear interpolation used to calculate the located cut. In this experiment, the areas estimated from the vertex coverage segmentations are on average 0.06% smaller than the true area. This error, however, is very small compared to the discretization errors of the crisp segmentation.

7. Conclusions

We have presented a framework for extending the concepts of fuzzy segmentation and partial belongingness to objects represented on an arbitrary graph. A key concept in this framework is the notion of a located cut, i.e., a graph cut specified with sub-edge precision.

Using the scheme presented in Section 5, it is possible to compute a located cut starting from any fuzzy segmentation of the graph vertices. Many popular graph-based segmentation methods produce this kind of fuzzy segmentation as output. Thus, practitioners can pick their favorite graph-based segmentation method, and augment it with the framework presented here. In most cases, the cost of computing a located cut, and subsequent vertex coverage segmentation, will be very small compared to the computational cost of the underlying segmentation method.

The experiments performed in Section 6 show that the proposed framework can be used to obtain segmentations from which feature measurements can be made with significantly improved precision. We emphasize that although the experiments were performed on 2D images and an 8-connected lattice, the proposed framework is defined for general graphs and thus applicable in a wide range of situations. A 3D segmentation result is shown in Fig. 10.

Acknowledgements

The authors wish to thank Dr Robin Strand at the Centre for Image Analysis, Uppsala University, Sweden, for valuable discussions during the writing of this manuscript. They would also like to thank the anonymous reviewers for the detailed suggestions that significantly improved the exposition. J. Lindblad and N. Sladoje are financially supported by the Ministry of Science of the Republic of Serbia through the Projects ON144018 and ON144029.

References

[1] R. Strand, Distance functions and image processing on point-lattices, Ph.D. thesis, Uppsala University, 2008.
[2] L. Grady, Space-variant machine vision—a graph theoretic approach, Ph.D. thesis, Boston University, 2004.
[3] L. Grady, M.-P. Jolly, Weights and topology: a study of the effects of graph construction on 3D image segmentation, in: D. Metaxas, et al. (Eds.), Proceedings of MICCAI 2008, vol. 1, Springer-Verlag, 2008, pp. 153–161.
[4] N. Sladoje, On analysis of discrete spatial fuzzy sets in 2 and 3 dimensions, Ph.D. thesis, Swedish University of Agricultural Sciences, 2005.
[5] N. Sladoje, J. Lindblad, Estimation of moments of digitized objects with fuzzy borders, in: Proceedings of ICIAP 2005, pp. 188–195.
[6] N. Sladoje, J. Lindblad, Pixel coverage segmentation for improved feature estimation, in: P. Foggia, et al. (Eds.), Proceedings of the 15th International Conference on Image Analysis and Processing, ICIAP, in: LNCS, vol. 5716, Springer-Verlag, 2009, pp. 929–938.
[7] N. Sladoje, J. Lindblad, High-precision boundary length estimation by utilizing gray-level information, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2009) 357–363.
[8] F. Malmberg, J. Lindblad, I. Nyström, Sub-pixel segmentation with the image foresting transform, in: P. Wiederhold, R.P. Barneva (Eds.), Proceedings of IWCIA 2009, vol. 5852, Springer, 2009, pp. 201–211.
[9] Y. Boykov, G. Funka-Lea, Graph cuts and efficient N–D image segmentation, International Journal of Computer Vision 70 (2006) 109–131.
[10] J. Shi, J. Malik, Normalized cuts and image segmentation, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.97CB36082), pp. 731–737.
[11] A.X. Falcão, J. Stolfi, R.A. Lotufo, The image foresting transform: theory, algorithms, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 19–29.
[12] A.X. Falcão, F.P. Bergo, Interactive volume segmentation with differential image foresting transforms, IEEE Transactions on Medical Imaging 23 (2004) 1100–1108.
[13] K. Ciesielski, J. Udupa, P. Saha, Y. Zhuge, Iterative relative fuzzy connectedness for multiple objects with multiple seeds, Computer Vision and Image Understanding 107 (2007) 160–182.
[14] F. Malmberg, I. Nyström, A. Mehnert, C. Engstrom, E. Bengtsson, Relaxed image foresting transforms for interactive volume image segmentation, in: B.M. Dawant, D.R. Haynor (Eds.), Proceedings of SPIE Medical Imaging, vol. 7623, 762340.
[15] L. Grady, Random walks for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1768–1783.
[16] J. Cousty, G. Bertrand, L. Najman, M. Couprie, Watershed cuts: thinnings, shortest path forests, and topological watersheds, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 925–939.
[17] L. Grady, C. Alvino, The piecewise smooth Mumford–Shah functional on an arbitrary graph, IEEE Transactions on Image Processing 18 (2009) 2547–2561.
[18] A.K. Sinop, L. Grady, A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm, in: Proc. of ICCV 2007, IEEE Computer Society, IEEE, 2007.
[19] C. Couprie, L. Grady, L. Najman, H. Talbot, Power watersheds: a new image segmentation framework extending graph cuts, random walker and optimal forest, in: Proceedings of International Conference on Computer Vision, ICCV, IEEE, 2009.
[20] P.A. Miranda, A.X. Falcão, Links between image segmentation based on optimum-path forest and minimum cut in graph, Journal of Mathematical Imaging and Vision 35 (2009) 128–142.
[21] G.M. Foody, R.M. Lucas, P.H. Curran, M. Honzak, Non-linear mixture modeling without end-members using an artificial neural network, International Journal of Remote Sensing 18 (1997) 937–953.
[22] J. Jua, E. Kolaczykb, S. Gopa, Gaussian mixture discriminant analysis and sub-pixel land cover characterization in remote sensing, Remote Sensing and Environment 84 (2003) 550–560.
[23] S. Verbeiren, H. Eerens, I. Piccard, J. Bauwens, I. Van Orshoven, Sub-pixel classification of spot-vegetation time series for the assessment of regional crop areas in Belgium, International Journal of Applied Earth Observation and Geoinformation 10 (2008) 486–497.
[24] K.V. Leemput, F. Maes, D. Vandermeulen, P. Suetens, A unifying framework for partial volume segmentation of brain MR images, IEEE Transactions on Medical Imaging 22 (2003) 105–119.
[25] K. Vincken, A. Koster, M. Viergever, Probabilistic segmentation of partial volume voxels, Pattern Recognition Letters 15 (1994) 477–484.
[26] N. Bousion, M. Hatt, A. Reilhac, D. Visvikis, Fully automated partial volume correction in PET based on a wavelet approach without the use of anatomical information, in: Proceedings of IEEE Nuclear Science Symposium, IEEE Society, 2007, pp. 2812–2816.
[27] H. Choi, D. Haynor, Y. Kim, Partial volume tissue classification of multichannel magnetic resonance images—a mixed model, IEEE Transactions on Medical Imaging 10 (1991) 395–407.
[28] A. Souza, J.K. Udupa, P.K. Saha, Volume rendering in the presence of partial volume effects, IEEE Transactions on Medical Imaging 24 (2005) 223–235.
[29] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[30] J. Lindblad, Surface area estimation of digitized 3D objects using weighted local configurations, Image and Vision Computing 23 (2005) 111–122.
[31] R. Klette, J. Žunić, Multigrid convergence of calculated features in image analysis, Journal of Mathematical Imaging and Vision 13 (2000) 173–191.